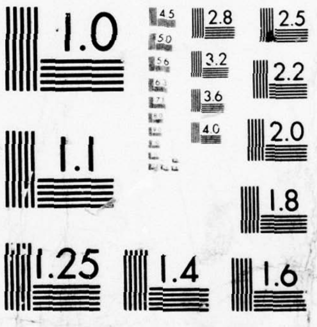
AD-A058 601   STANFORD UNIV CALIF DEPT OF COMPUTER SCIENCE   F/G 9/2
PROVING TERMINATION WITH MULTISET ORDERINGS.(U)
MAR 78   N DERSHOWITZ, Z MANNA   MDA903-76-C-0206
UNCLASSIFIED   STAN-CS-78-651   NL

1 OF 1
AD
A058601

END
DATE
FILMED
11-78
DDC


1.0
4.5
5.0
5.6
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

Stanford Artificial Intelligence Laboratory
Memo AIM-310

(12) (11) Mar[redacted]978

Computer Science Department
Report No. STAN-CS-78-651, (14) AIM-310

LEVEL II

(6) PROVING TERMINATION WITH MULTISET ORDERINGS.

[redacted]

(10) Nachum Dershowitz [redacted] Zohar Manna

(12) 35p.

(15) MDA903-76-C-0206, AFOSR-76-2909

Research sponsored by

United States Air Force

National Science Foundation

Advanced Research Projects Agency
ARPA Order No. 2494

(9) Technical rept.,

COMPUTER SCIENCE DEPARTMENT
Stanford University

LELAND STANFORD JUNIOR UNIVERSITY
ORGANIZED 1891

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER STAN-CS-78-651, AIM-310	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Proving Termination with Multiset Orderings		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER AIM-310
7. AUTHOR(s) Nachum Dershowitz and Zohar Manna		8. CONTRACT OR GRANT NUMBER(s) MDA903-76-C-0206 (ARPA) AFOSR-76-2909 (Air Force) MCS-76-83655 (NSF)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory Stanford University Stanford, CA 94305		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order 2494
11. CONTROLLING OFFICE NAME AND ADDRESS Eugene Stubbs ARPA/PM 1400 Wilson Blvd., Arlington, VA 22209		12. REPORT DATE March 1978
		13. NUMBER OF PAGES 33
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Philip Surra, ONR Representative Durand Aeronautics Building Room 165 Stanford University Stanford, CA 94305		15. SECURITY CLASS. (of this report) 15
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Releasable without limitations on dissemination

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A common tool for proving the termination of programs is the well-founded set, a set ordered in such a way as to admit no infinite descending sequences. The basic approach is to find a termination function that maps the elements of the program into some well-founded set, such that the value of the termination function is continually reduced throughout the computation All too often, the termination functions required are difficult to find and are of a complexity out of proportion to the program under consideration.

Next page

Block 20

However, by providing more sophisticated well-founded sets, the corresponding termination functions can be simplified.

Given a well-founded set S, we consider multisets over S, "sets" that admit multiple occurrences of elements taken from S. We define an ordering on all finite multisets over S that is induced by the given ordering on S. This multiset ordering is shown to be well-founded.

The value of the multiset ordering is that it permits the use of relatively simple and intuitive termination functions in otherwise difficult termination proofs. In particular, we apply the multiset ordering to provide simple proofs of the termination of production systems, programs defined in terms of sets of rewriting rules.

Stanford Artificial Intelligence Laboratory
Memo AIM-310

March 1978

Computer Science Department
Report No. STAN-CS-78-651

PROVING TERMINATION WITH MULTISET ORDERINGS

by

Nachum Dershowitz and Zohar Manna

ABSTRACT

A common tool for proving the termination of programs is the *well-founded set*, a set ordered in such a way as to admit no infinite descending sequences. The basic approach is to find a *termination function* that maps the elements of the program into some well-founded set, such that the value of the termination function is continually reduced throughout the computation. All too often, the termination functions required are difficult to find and are of a complexity out of proportion to the program under consideration. However, by providing more sophisticated well-founded sets, the corresponding termination functions can be simplified.

Given a well-founded set S, we consider *multisets* over S, "sets" that admit multiple occurrences of elements taken from S. We define an ordering on all finite multisets over S that is induced by the given ordering on S. This *multiset ordering* is shown to be well-founded.

The value of the multiset ordering is that it permits the use of relatively simple and intuitive termination functions in otherwise difficult termination proofs. In particular, we apply the multiset ordering to provide simple proofs of the termination of *production systems*, programs defined in terms of sets of rewriting rules.

The authors are also affiliated with the Department of Applied Mathematics of the Weizmann Institute of Science, Rehovot, Israel.

This research was supported in part by the United States Air Force Office of Scientific Research under Grant AFOSR-76-2909 (sponsored by the Rome Air Development Center, Griffiss AFB, NY), by the National Science Foundation under Grant MCS 76-83655, and by the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-76-C-0206.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Stanford University or the U. S. Government.

Reproduced in the U.S.A. Available from the National Technical Information Service, Springfield, Virginia 22161.

I. INTRODUCTION

The use of well-founded sets for proving that programs terminate has been suggested by Floyd [1967]. A *well-founded set* $(S, \succ)$ consists of a set of elements S and an ordering $\succ$ defined on the elements, such that there can be no infinite decreasing sequences of elements. The idea is to find a well-founded set and a *termination function* that maps the elements of the program into that set, such that the value of the termination function is continually reduced throughout the computation. Since, by the nature of the set, that value cannot decrease indefinitely, the program must terminate. The well-founded sets most frequently used for this purpose are the natural numbers under the "greater-than" ordering and n-tuples of natural numbers under the lexicographic ordering.

In this paper, we define and illustrate a class of orderings on multisets. *Multisets*, sometimes called *bags*, are like sets, but allow multiple occurrences of identical elements. For example, $\{3, 3, 3, 4, 0, 0\}$ is a multiset of natural numbers; it is identical to the multiset $\{0, 3, 3, 0, 4, 3\}$, but is distinct from $\{3, 4, 0\}$.

The ordering $\succ$ on any given well-founded set S can be extended to form a well-founded ordering $\gg$ on the finite multisets over S. In this ordering, $M \gg M'$, for two finite multisets M and M' over S, if M' may be obtained from M by the removal of at least one element from M and/or by the replacement of one or more elements in M with any finite number of elements taken from S, each of which is smaller than one of the replaced elements. Thus, if S is the set of natural numbers $0, 1, 2, \ldots$ with the $>$ ordering, then under the corresponding *multiset ordering* $\gg$ *over* S, the multiset $\{3, 3, 4, 0\}$ is greater than each of the three multisets $\{3, 4\}$, $\{3, 2, 2, 1, 1, 1, 4, 0\}$, and $\{3, 3, 3, 3, 2, 2\}$. In the first case, two elements have been removed; in the second case, an occurrence of 3 has been replaced by two occurrences of 2 and three occurrences of 1; and in the third case, the element 4 has been replaced by two occurrences each of 3 and 2, and in addition the element 0 has been removed.

As an example of the use of a multiset ordering for a proof of termination, consider the following trivial program to empty a shunting yard of all trains:

```
loop until the shunting yard is empty
     select a train
     if the train consists of only a single car
        then remove it from the yard
        else split it into two shorter trains
        fi
     repeat .
```

This program is nondeterministic, as it does not indicate which train is to be selected nor how the train is to be split.

Let Y denote the set of trains in the yard, and $trains(Y)$ be the number of trains in the yard. For any train $t \in Y$, let $cars(t)$ be the number of cars it contains. We present two proofs of termination.

If we take the set of natural numbers as our well-founded set, then we are led to the selection of the termination function

$$\tau(Y) = 2 \cdot \sum_{t \in Y} cars(t) - trains(Y)$$

(see Dijkstra [1976]). This solution uses the fact that "splitting" conserves the number of cars in the yard, $\Sigma\, cars(t)$. Thus, splitting a train increases the number of trains in the yard, $trains(Y)$, by 1, thereby decreasing the current value of the termination function τ by 1. Removing a one-car train from the yard reduces $2 \cdot \Sigma\, cars(t)$ by 2 and increases $-trains(Y)$ by 1, thereby decreasing τ by 1.

If we use multisets of natural numbers as our well-founded set, then the function

$$\tau(Y) = \{cars(t) : t \in Y\}$$

demonstrates the termination of the shunting program. That is, for any configuration of the yard Y, $\tau(Y)$ denotes the multiset containing the size of each of the trains in Y. Each iteration of the program loop clearly decreases the value of $\tau(Y)$ under the multiset ordering: removing a train from the yard reduces the multiset by removing one element; splitting a train replaces one element with two smaller ones, corresponding to the two shorter trains.

Programs are sometimes written in the form of a *production system*. The following system of three rewrite rules is an example:

$$\boxed{\begin{array}{lcl} white,\ red & \Rightarrow & red,\ white \\ blue,\ red & \Rightarrow & red,\ blue \\ blue,\ white & \Rightarrow & white,\ blue\ . \end{array}}$$

This program solves the "Dutch-flag" problem: Assuming that we have a series of marbles, colored *red*, *white* or *blue* and placed side by side in no particular order, then the above program will rearrange the marbles so that all the *red* marbles are on the left, all *blue* marbles are on the right, and all *white* marbles are in the middle. The first rule, for example, states that if anywhere in the series there is an adjacent pair of marbles, the left one *white* and the right one *red*, then they should be exchanged so that the *red* marble is on the left and the *white* one is on the right.

The three rules may be applied in any order and to any pair of marbles matching a left-hand side of a rule. The program terminates when no rule can be applied. Clearly, if no rule can be applied, the marbles are in the desired order, since nowhere does a *red* marble have anything but a *red* marble to its immediate left (or else one of the first two rules could be applied), and nowhere does a *blue* marble have anything but a *blue* marble to its right (or else one of the last two rules could be applied). The only thing we need to ascertain is that the program will not just keep on running, never reaching a situation when no rule can be applied; in other words, we must prove that the above production system terminates.

There are several ways of proving termination. The three we give here all use the following ordering on colors:

blue is greater than *white* and *white* is greater than *red* .

It follows from the transitivity of orderings that *blue* is also greater than *red* .

The first method counts the total number of "inversions" of marbles, i.e. the number of pairs of marbles a and b (not necessarily adjacent), such that a appears to the left of b and the color of a is greater than the color of b . For example, if five marbles are arranged *blue*, *red*, *white*, *red*, *blue* , then there are four inversions: *blue-red* , *blue-white* , *blue-red* , and *white-red* . Thus, the well-founded set is the set of natural numbers under their standard $>$ ordering, and the termination function counts the number of inversions by summing, for each marble, the number of marbles with a greater color to its left. Each of the rules, when applied, eliminates one inversion by exchanging the positions of one inverted pair, thereby reducing the value of the termination function by one.

For the second method, suppose that there are n marbles. The well-founded set we use is the set of n-tuples of colors. This tuple is ordered *lexicographically*: it is reduced if some component is reduced without changing any component to its left. The termination function simply yields the tuple containing the colors of the marbles in order, from left to right. To prove termination, we note that whenever one of the rules is applied to two marbles, only the values of the two corresponding components of the tuple change. By the nature of the lexicographic ordering, we need only consider the change in the left component, and indeed it is reduced in its color: if it was *blue* , then now it is either *white* or *red* , and if it was *white* , then now it is *red* .

The third solution illustrates the use of multiset orderings. Each of the n positions in the series is assigned a number, beginning with n-1 at the left, and going down to 0 for the rightmost position. We take the multisets of pairs of the form (*position*, *color*) as the well-founded set. The position-color pairs are ordered lexicographically: we say that a pair is greater than another, if it has a higher position number than the other, or if it has the same position number but a greater color. For each marble, the termination function yields one pair, giving its position and color. When a rule is applied to the marbles at positions i and $i-1$, it decreases the value of the multiset by decreasing the color of the marble at position i. The fact that the color at position $i-1$ is increased does not matter, since any pair with position i is lexicographically greater than any pair with position $i-1$, regardless of the colors.

These two examples have demonstrated how the multiset ordering may be used in termination proofs. These proofs, however, did not have a clear advantage over the alternative proofs, using the more common "greater-than" relation on the natural numbers and lexicographic ordering on n-tuples. In practice using these conventional orderings often leads to complex termination functions that are difficult to discover. For example, the termination proofs of programs involving stacks and production systems are often quite complicated and require much more subtle orderings and termination functions. Finding an appropriate ordering and termination function for such programs is a well-known challenge among researchers in the field of program verification. In the remainder of this paper, we shall demonstrate how the multiset ordering permits the use of relatively simple and intuitive termination functions in otherwise difficult termination proofs.

In the next section, Section II, we rigorously define the multiset ordering and prove that it is well-founded. In Section III, we apply the multiset ordering to a number of termination proofs of programs. Then, in Section IV, we use the multiset ordering to prove the termination of production systems.

II. THE MULTISET ORDERING

A *partially-ordered set* $(S, >)$ consists of a set S and a transitive and irreflexive binary relation $>$ on elements of S. For example, both the set $\mathbb{Z}$ of all integers and the set $\mathbb{N}$ of nonnegative integers are ordered by the "greater-than" relation $>$. The ordering may be partial: for two distinct elements a and b of the set, we may have neither $a>b$ nor $b>a$.

A partially-ordered set is said to be *well-founded* if there can be no infinite decreasing sequences of elements from the set. Thus, the set $(\mathbb{N}, >)$ is well-founded, since any descending sequence of natural numbers cannot go beyond 0. On the other hand, the partially-ordered set $(\mathbb{Z}, >)$ is not well-founded.

For a given partially-ordered set $(S, >)$, we consider the *multisets* over S, i.e. unordered collections of elements ("sets") that may have multiple occurrences of identical elements. We denote by $\mathcal{M}(S)$ the set of all finite multisets with elements taken from the set S, and associate an ordering $\gg$ on $\mathcal{M}(S)$ that is induced by the given ordering $>$ on S.

In the following definition, as well as in the rest of this paper, set operators will denote their multiset analogues: The equality $A=B$ of two multisets, for example, means that any element occurring exactly n times in A, also occurs exactly n times in B, and vice versa. The union of two multisets $A \cup B$ is a multiset containing $m+n$ occurrences of any element occurring m times in A and n times in B. For example, the union of the multisets $\{2, 2, 4\}$ and $\{2, 0, 0\}$ is $\{2, 2, 4, 2, 0, 0\}$.

For a partially-ordered set $(S, >)$, the *multiset ordering* $\gg$ on $\mathcal{M}(S)$ is defined as follows:

$$M \gg M'$$

if for some multisets X, Y, Z in $\mathcal{M}(S)$, where X is not empty,

$$M = X \cup Z \quad \text{and} \quad M' = Y \cup Z$$

and

$$(\forall y \in Y)(\exists x \in X)\ x > y \ .$$

In words, a multiset is reduced by the removal of at least one element (those in X) and their replacement with any finite number — possibly zero — of elements (those in Y), each of which is smaller than one of the elements that have been removed.

We must first show that $\gg$ is in fact a partial ordering, i.e. that if $>$ is irreflexive and transitive, then $\gg$ also is:

- To show irreflexivity, we must show that there can be no multiset M such that $M \gg M$. Suppose that $M \gg M$, then there would be some nonempty finite multiset X, such that $M = X \cup Z$ and $(\forall y \in X)(\exists x \in X) x > y$. In other words, for every element of X there would be a distinct element of X greater than it, which is impossible for a finite X.
- To show transitivity of $\gg$, consider the following irreflexive relation $\gg'$ on multisets in $\mathcal{M}(S)$: $Z \cup \{x\} \gg' Z \cup Y$ if $(\forall y \in Y) x > y$. In other words, a finite multiset is reduced in the relation $\gg'$ by replacing a *single* element with zero or more smaller elements. Note that the multiset ordering $\gg$ is the transitive closure of the relation $\gg'$, i.e. $M \gg M'$ if and only if M' can be obtained from M by replacing elements in M one by one. It follows that $\gg$ is transitive.

We have the

THEOREM: *The multiset ordering* $(\mathcal{M}(S), \gg)$ *over* $(S, >)$ *is well-founded, if and only if* $(S, >)$ *is.*

Proof:

- "only if" part. If $(S, >)$ is not well-founded, then there exists an infinite decreasing sequence $s_1 > s_2 > s_3 > \ldots$ of elements in S. The corresponding sequence of singletons $\{s_1\} \gg \{s_2\} \gg \{s_3\} \gg \ldots$ forms an infinite decreasing sequence of elements in $\mathcal{M}(S)$, and $(\mathcal{M}(S), \gg)$ is therefore not well-founded.
- "if" part. Assume that $(S, >)$ is well-founded. We first extend S by adding to it an element $\perp$, and extend the ordering $>$ on S to make $\perp$ the least element, i.e. for every element $s \neq \perp$ in S, $s > \perp$. Clearly S remains well-founded, thereby. Now, suppose that $(\mathcal{M}(S), \gg)$ is not well-founded; therefore, there exists an infinite decreasing sequence $M_1 \gg M_2 \gg M_3 \gg \ldots$ of multisets from $\mathcal{M}(S)$. We derive a contradiction by constructing the following tree. Each node in the tree is labelled with some element of S; at each stage of the construction, the set of all terminal nodes in the tree forms a multiset in $\mathcal{M}(S)$.

> Begin with a root node with children corresponding to each element of M_1. Then since $M_1 \gg M_2$, there must exist multisets X, Y, Z, such that $M_1 = X \cup Z$, $M_2 = Y \cup Z$, X is not empty, and $(\forall y \in Y)(\exists x \in X) x > y$. Then for each $y \in Y$, add a son labelled y to the corresponding x. In addition, grow a child $\perp$ from each of the elements of X. (Since X is nonempty, growing $\perp$ ensures that even if Y is empty, at least one node is added to the tree. Since Y is finite, the nodes corresponding to X each have a finite number of sons.) Repeat the process for $M_2 \gg M_3$, $M_3 \gg M_4$, and so on.

Since at least one node is added to the tree for each multiset M_i in the sequence, were the sequence infinite, the tree corresponding to the sequence would also be. But by Konig's Infinity Lemma, an infinite tree (with a finite number of children for each node) must have an infinite path. On the other hand, by our construction, all paths in the tree are descending in the well-founded ordering $>$ on S, and must be finite. Thus, we have derived a contradiction, implying that the sequence $M_1, M_2, M_3, \ldots$ cannot be infinite. □

Remark: If $(S, >)$ is totally ordered, then for any two multisets $M, M' \in \mathcal{M}(S)$, one may decide whether $M \gg M'$ by first sorting the elements of both M and M' in descending order (with respect to the relation $>$) and then comparing the two sorted sequences lexicographically. For example, to compare the multisets $\{3, 3, 4, 0\}$ and $\{3, 2, 1, 2, 0, 4\}$, one may compare the sorted sequences $(4, 3, 3, 0)$ and $(4, 3, 2, 2, 1, 0)$. Since $(4, 3, 3, 0)$ is lexicographically greater than $(4, 3, 2, 2, 1, 0)$, it follows that $\{3, 3, 4, 0\} \gg \{3, 2, 1, 2, 0, 4\}$. □

Remark: *If* $(S, >)$ *is of order type* α, *then the multiset ordering* $(\mathcal{M}(S), \gg)$ *over* $(S, >)$ *is of order type* ω^α. This follows from the fact that there exists a mapping ψ from $\mathcal{M}(S)$ onto ω^α that is one-to-one and order-preserving, i.e. if $M \gg M'$ for $M, M' \in \mathcal{M}(S)$, then the ordinal $\psi(M)$ is greater than $\psi(M')$. That mapping is

$$\psi(M) = \sum_{m \in M} \omega^{\varphi(m)} ,$$

where Σ denotes the natural (i.e. commutative) sum of ordinals, and φ is the one-to-one order-preserving map from S onto α. □

Remark: Consider the special case where there is a bound k on the number of replacement elements, i.e. $|Y| < k$. *Any termination proof using this bounded multiset ordering over* $\mathbb{N}$ *may be translated into a proof using* $(\mathbb{N}, >)$. This may be done using the order-preserving function

$$\psi(M) = \sum_{n \in M} k^n$$

which maps multisets over the natural numbers into the natural numbers by summing the number k^n for every natural number n in a multiset M. Two special cases of interest are: if $|Y| \leq |X|$ (i.e. the size of the multiset is not increased), then the simpler function

$$\psi(M) = \sum_{n \in M} n + |M|$$

is order-preserving; if $|Y| = |X|$ (i.e. the size of the multiset is constant), then

$$\psi(M) = \sum_{n\in M} n$$

is order-preserving. □

We turn now to consider *nested multisets*, by which we mean that the elements of the multisets may belong to some base set S, or may be multisets of elements of S, or may be multisets containing both elements of S and multisets of elements of S, and so on. For example,

$$\{\{1, 1\}, \{\{0\}, 1, 2\}, 0\}$$

is a nested multiset. More formally, given a partially-ordered set $(S, \succ)$, a *nested multiset over* S is either an element of S, or else it is a finite multiset of nested multisets over S. We denote by $\mathcal{M}^*(S)$ the set of nested multisets over S.

We define now a *nested multiset ordering* $\gg^*$ on $\mathcal{M}^*(S)$; it is a recursive version of the standard multiset ordering. For two elements $M, M' \in \mathcal{M}^*(S)$, we say that

$$M \gg^* M'$$

if

- $M, M' \in S$ and $M \succ M'$

(two elements of the base set are compared using $\succ$),
or else

- $M \notin S$ and $M' \in S$

(any multiset is greater than any element of the base set),
or else

- $M, M' \notin S$, and for some $X, Y, Z \in \mathcal{M}^*(S)$, where X is not empty,

$$M = X \cup Z \quad \text{and} \quad M' = Y \cup Z$$

and

$$(\forall y \in Y)(\exists x \in X)\ x \gg^* y .$$

For example, the nested multiset

$$\{\{1, 1\}, \{\{0\}, 1, 2\}, 0\}$$

is greater than

$$\{\{1, 0, 0\}, 5, \{\{0\}, 1, 2\}, 0\} ,$$

since $\{1, 1\}$ is greater than both $\{1, 0, 0\}$ and 5. The nested multiset

$$\{\{1, 1\}, \{\{0\}, 1, 2\}, 0\}$$

is also greater than

$\{\{\{\}, 1, 2\}, \{5, 5, 2\}, 5\}$,

since $\{\{0\}, 1, 2\}$ is greater than each of the three elements $\{\{\}, 1, 2\}$, $\{5, 5, 2\}$, and 5 .

Let $\mathcal{M}^i(S)$ denote the set of all nested multisets of depth i . In other words, $\mathcal{M}^0(S)=S$ and $\mathcal{M}^{i+1}(S)$ contains the multisets whose elements are taken from $\mathcal{M}^0(S)$, $\mathcal{M}^1(S)$, . . . , $\mathcal{M}^i(S)$, with at least one element taken from $\mathcal{M}^i(S)$. Thus, the set $\mathcal{M}^*(S)$ is the infinite union of the disjoint sets $\mathcal{M}^0(S)$, $\mathcal{M}^1(S)$, $\mathcal{M}^2(S)$, ... The following property holds:

> *For two nested multisets,* M *and* M' , *if the depth of* M *is greater than the depth of* M' , *then* $M \gg^* M'$.

In other words, the elements of $\mathcal{M}^i(S)$ are all greater than the elements of $\mathcal{M}^j(S)$, for any $j<i$.

Proof: This property may be proved by induction on depth. It holds vacuously for M of depth 0 . For the inductive step, assume that nested multisets of depth i are greater than nested multisets of depth less than i ; we must show that a nested multiset M of depth $i+1$ is greater than any nested multiset M' of lesser depth. If the depth of M' is 0 , then $M' \in S$ while $M \notin S$, and therefore $M \gg^* M'$, as desired. If the depth of M' is less than i but greater than 0 , then each of the elements in M' is of depth less than $i-1$. The nested multiset M , on the other hand, is of depth $i+1$ and must therefore contain some element of depth i . By the inductive hypothesis, that element is greater than each of the elements in M' . Again, it follows that $M \gg^* M'$. □

The relation $\gg^*$ is a partial ordering; it can be shown to be both irreflexive and transitive. The following theorem gives the condition under which it is well-founded:

> THEOREM: *The nested multiset ordering* $(\mathcal{M}^*(S), \gg^*)$ *over* $(S, >)$ *is well-founded, if and only if* $(S, >)$ *is well-founded.*

Proof:

- ● "only if" part. If $(S, >)$ is not well-founded, then there exists an infinite decreasing sequence $s_1 > s_2 > s_3 > \ldots$ of elements in S . This sequence is also an infinite decreasing sequence of elements in $\mathcal{M}^*(S)$ under $\gg^*$, and $(\mathcal{M}^*(S), \gg^*)$ is therefore not well-founded.

● "if" part. In order to show that $(\mathcal{M}^*(S), \gg^*)$ is well-founded, it suffices to show that each $\mathcal{M}^i(S)$ is itself well-founded under $\gg^*$. For assume that $\mathcal{M}^*(S)$ were not well-founded, then there would exist an infinite decreasing sequence of nested multisets $M_1 \gg^* M_2 \gg^* \ldots$. By the above property, the depth of any nested multiset M_{i+1} in the sequence cannot be greater than the depth of its predecessor M_i. Since the sequence is infinite, it must have an infinite subsequence of nested multisets all of the same depth i, which contradicts the well-foundedness of $\mathcal{M}^i(S)$.

We prove that each $(\mathcal{M}^i(S), \gg^*)$ is well-founded by induction on i: The ordering $\gg^*$ on $\mathcal{M}^0(S)=S$ is simply the ordering $>$ on S, and it follows that $(\mathcal{M}^0(S), \gg^*)$ is well-founded. For the inductive step, assume that each $(\mathcal{M}^j(S), \gg^*)$, $j<i$, is well-founded, and note that each of the elements of $\mathcal{M}^i(S)$ is a member of the union of $\mathcal{M}^0(S), \mathcal{M}^1(S), \ldots, \mathcal{M}^{i-1}(S)$. By the induction hypothesis, each of these $\mathcal{M}^j(S)$ is well-founded under $\gg^*$; therefore their union under $\gg^*$ also is. Furthermore, the ordering $\gg^*$ on two nested multisets from $\mathcal{M}^i(S)$ is exactly the standard multiset ordering over that union, and since the union is well-founded, so is $\mathcal{M}^i(S)$. □

Remark: We have seen above that for $(S, >)$ of order type α, the multiset ordering $(\mathcal{M}(S), \gg)$ is of order type ω^α. In a similar manner, it can be shown that the order type of $(\mathcal{M}^i(S), \gg^*)$ is

$$\left.\omega^{\omega^{\cdot^{\cdot^{\cdot^{\omega^{\alpha}}}}}}\right\} i \textit{ times} ,$$

the limit of which is the ordinal ϵ_0 — provided that α is less than ϵ_0. Thus, *if* $(S, >)$ *is of order type less than* ϵ_0, *then* $(\mathcal{M}^*(S), \gg^*)$ *is of order type* ϵ_0. (Gentzen [1938] used an ϵ_0 ordering to prove the termination of his normalization procedure for proofs in arithmetic.) □

In the following two sections, we shall apply the multiset ordering to problems of termination, first proving the termination of programs, and then proving the termination of production systems.

III. TERMINATION OF PROGRAMS

The following basic theorem is generally used to prove the termination of programs:

> THEOREM (Floyd): *A program P with variables $\bar{x}$ ranging over a domain D terminates, if and only if there exist*
>
> *a set of labels $\mathcal{L}$ cutting all the loops in P,*
>
> *a well-founded set $(W, \succ)$, and*
>
> *a termination function τ mapping $\mathcal{L}\times D$ into W,*
>
> *such that whenever control traverses a path from one label to another, the value of the termination function $\tau_L(\bar{x})$ decreases for the current label L and value of $\bar{x}$.*

Proof:

- "only if" part. If the program does terminate, then the set $(\mathcal{L}\times D, \succ_P)$ is well-founded, where the relation $\succ_P$ is defined so that $(L, \bar{d})\succ_P(L', \bar{d}')$ if the program reaches the label L with the value $\bar{d}$ before it reaches L' with the value $\bar{d}'$. Thus, if $\tau_L(\bar{x})$ returns the pair $(L, \bar{x})$, then with each traversal of a path, the current value of $\tau_L(\bar{x})$ decreases.
- "if" part. If the program does not terminate, then there exists an infinite sequence of label-value pairs $(L_1, \bar{d}_1), (L_2, \bar{d}_2), \ldots$, corresponding to the sequence of labels through which control passes during a nonterminating computation and the values of the variables at those points. Since the function τ decreases with each traversal of a path, it follows that the sequence $\tau_{L_1}(\bar{d}_1), \tau_{L_2}(\bar{d}_2), \ldots$ forms an infinite decreasing sequence in the set W, contradicting its well-foundedness.

In the following examples, we shall prove the termination of programs using multiset orderings as the well-founded set.

EXAMPLE 1: *Counting tips of binary trees.*

Consider a simple program to count the number of tips — terminal nodes (without descendents) — in a full binary tree. Each tree y that is not a tip has two subtrees, $left(y)$ and $right(y)$. The program is

```
$S$ := $(t)$
$c$ := 0
loop until $S$=( )
    $y$ := $s_1$
    if $tip(y)$ then $S$ := $(s_{|S|}, \ldots, s_2)$
                     $c$ := $c+1$
                else $S$ := $(s_{|S|}, \ldots, s_2, left(y), right(y))$
                fi
    repeat .
```

It employs a stack S with the $|S|$ elements $s_{|S|}, \ldots, s_2, s_1$, and terminates when S is empty. At that point, the variable c is to contain the total number of tip nodes in the given tree t.

Initially the given tree is placed in the stack. With each iteration the subtree at the top of the stack is tested to determine whether it is a tip: if it is, then it is removed from the stack and the count is incremented by 1; if it is not a tip, then it is replaced in the stack with its two subtrees, so that the number of tips in each subtree may be counted.

The termination of this program may be proved using the well-founded set $(\mathbb{N}, >)$. The appropriate termination function is

$$\tau(S) = \sum_{s \in S} nodes(s) ,$$

where $nodes(s)$ is the total number of nodes in the subtree s — not just the tip nodes. To show that the value of τ decreases with each loop iteration, we must consider two cases: If the subtree s_1 is a tip node, then that node is removed from the stack, and the sum is decreased by 1. If s_1 is not a tip, then it is replaced by its two subtrees, $left(s_1)$ and $right(s_1)$. But s_1 contains one node more than $left(s_1)$ and $right(s_1)$ combined, and again the sum is reduced.

Using the multiset ordering over trees, we can prove termination with the simple termination function

$$\tau(S) = \{s : s \in S\} ,$$

where the trees themselves are ordered by the natural well-founded subtree ordering, i.e. any tree is greater than its subtrees. Thus, removal of a tree from the stack decreases τ in the multiset ordering by removing an element, and the replacement of a tree with two smaller subtrees decreases τ.

This solution uses multisets over trees. One could just as well have used multisets over natural numbers, taking as the termination function $\{nodes(s):s\in S\}$ or $\{tips(s):s\in S\}$. The first solution, using the conventional well-founded set $(\mathbf{N}, >)$, does not provide such flexibility. □

EXAMPLE 2: *McCarthy's 91-function.*

The following is a contrived program to compute the simple function

$$f(x) = \mathit{if}\ x>100\ \mathit{then}\ x-10\ \mathit{else}\ 91$$

over the set of integers $\mathbf{Z}$, in a round-about manner. Though this program is short, the proof of its correctness and termination are nontrivial, and for this reason it is often used to illustrate proof methods.

The program is:

```
n := 1
z := x
loop L: assert  $f(x)=f^n(z)$,  $n\geq 1$
     if  $z>100$  then  $n := n-1$
                       $z := z-10$
                  else  $n := n+1$
                        $z := z+11$
                  fi
     until  $n=0$ .
     repeat
assert  $z=f(x)$ .
```

The predicates $f(x)=f^n(z)$ and $n\geq 1$, in the **assert** clause at the head of the loop, are invariant assertions; they hold whenever control is at label L. When the program terminates, the variable z contains the value of $f(x)$, since the loop is exited when the condition $n=0$ of the **until** clause is satisfied; at that point, $f(x)=f^0(z)=z$.

Using the conventional well-founded set $(\mathbf{N}, >)$, Katz and Manna [1975] prove the termination of this program with the termination function

$$\tau(n, z) = -2\cdot z+21\cdot n+2\cdot max(111, x)$$

at L.

For an alternative proof of termination, we consider the following well-founded partial-ordering $\succ$ on the integers:

$$a \succ b \quad \text{if and only if} \quad a < b \leq 111 .$$

(This is the same ordering on integers as in the usual structural-induction proof, due to Rod Burstall, of the recursive version of this program.) As the well-founded set, we use the set $(\mathcal{M}(\mathbb{Z}), \gg)$ of all multisets of integers, under the corresponding multiset ordering. The appropriate termination function τ at L yields a multiset in $\mathcal{M}(\mathbb{Z})$, and is defined as

$$\tau(n, z) = \{z, \ f(z), \ \ldots, \ f^{n-1}(z)\} .$$

We must show that for each loop iteration this function decreases. There are three cases to consider:

1) $z>100$ at L : In this case, the **then** branch of the conditional is executed and both n and z are decremented. When control returns to L (assuming that the loop has not been exited), we have, in terms of the old values of n and z ,

$$\tau(n-1, z-10) = \{z-10, \ f(z-10), \ \ldots, \ f^{n-2}(z-10)\} .$$

Since $z>100$, we have $f(z)=z-10$, and therefore

$$\tau(n-1, z-10) = \{f(z), \ f^2(z), \ \ldots, \ f^{n-1}(z)\} .$$

Thus, the value of the termination function τ has been decreased by removing the element z from the original multiset $\{z, f(z), \ldots, f^{n-1}(z)\}$.

2) $90 \leq z \leq 100$ at L : In this case, the **else** branch is taken and both n and z are incremented, yielding

$$\tau(n+1, z+11) = \{z+11, \ f(z+11), \ f^2(z+11), \ \ldots, \ f^{n}(z+11)\} .$$

Since $z+11>100$, we have $f(z+11)=z+1$ and $f^2(z+11)=f(z+1)$. Furthermore, either $z+1=101$ or else $z+1 \leq 100$, and in both case $f(z+1)=91=f(z)$ and consequently $f^2(z+11)=f(z)$. Thus, we get

$$\tau(n+1, z+11) = \{z+11, \ z+1, \ f(z), \ \ldots, \ f^{n-1}(z)\} .$$

Since $z<z+1<z+11 \leq 111$, we have $z \succ z+11$ and $z \succ z+1$. Accordingly, the multiset has been reduced by replacing the element z with the two smaller elements, $z+11$ and $z+1$.

3) $z \leq 89$ at L : The **else** branch is taken and we have

$$\tau(n+1,\ z+11) = \{z+11,\ f(z+11),\ f^2(z+11),\ \ldots,\ f^n(z+11)\} .$$

Since $z+11 \leq 100$, we have $f(z+11)=91$ and $f^2(z+11)=f(91)=91=f(z)$, and thus

$$\tau(n+1,\ z+11) = \{z+11,\ 91,\ f(z),\ \ldots,\ f^{n-1}(z)\} .$$

Again z has been replaced by two smaller elements (under the $\succ$ relation), $z+11$ and 91 . □

EXAMPLE 3: *Ackermann's function.*

Ackermann's function $a(m,\ n)$ over pairs of natural numbers is defined recursively as

```
a(m, n) ⇐ if  m=0  then  n+1
             else  if  n=0  then  a(m-1, 1)
                             else  a(m-1, a(m, n-1))
                             fi fi .
```

The following iterative program computes this function:

```
S  :=  (m)
z  :=  n
loop L:  assert  $a(m, n) = a(s_{|S|}, a(s_{|S|-1}, \ldots, a(s_2, a(s_1, z))\ldots))$
    if  $s_1=0$  then  $S := (s_{|S|}, \ldots, s_2)$
                       $z := z+1$
    else
    if  $z=0$  then  $S := (s_{|S|}, \ldots, s_2, s_1-1)$
                       $z := 1$
           else  $S := (s_{|S|}, \ldots, s_2, s_1-1, s_1)$
                       $z := z-1$
           fi fi
    until  $S=(\,)$
    repeat
assert  $z = a(m, n)$ .
```

The three branches of the conditional statement correspond to the three cases in the recursive program.

The termination of this program was proved by Manna and Waldinger [1978] using the intermittent-assertion technique. We give here two proofs using multisets.

● *Solution* 1 : Consider the set $\mathbb{N}\times\mathbb{N}$ of lexicographically-ordered pairs of natural numbers, and use the corresponding multiset ordering over $\mathbb{N}\times\mathbb{N}$. The termination function at L is

$$\tau(S,\ z)\ =\ \{(s_{|S|}+1,\ 0),\ \ (s_{|S|-1}+1,\ 0),\ \ \ldots,\ \ (s_2+1,\ 0),\ \ (s_1,\ z)\}\ .$$

Thus, $\tau(S,\ z)$ yields a multiset containing one pair per element in the stack S . Note that at L , the stack S is nonempty, and all the elements in S as well as z are nonnegative.

The proof considers three cases, corresponding to the three branches of the conditional in the loop:

1) $s_1=0$. If the loop is not exited, then the new value of τ at L is

$$\tau((s_{|S|},\ \ldots,\ s_2),\ z+1)\ =\ \{(s_{|S|}+1,\ 0),\ \ \ldots,\ \ (s_3+1,\ 0),\ \ (s_2,\ z+1)\}\ .$$

This represents a decrease in τ under the multiset ordering, since the element $(s_1,\ z)$ has been removed and the element $(s_2+1,\ 0)$ has been replaced by the smaller $(s_2,\ z+1)$.

2) $s_1\neq 0$ and $z=0$. In this case we obtain

$$\tau((s_{|S|},\ \ldots,\ s_2,\ s_1-1),\ 1)\ =\ \{(s_{|S|}+1,\ 0),\ \ \ldots,\ \ (s_2+1,\ 0),\ \ (s_1-1,\ 1)\}\ .$$

Thus, the element $(s_1,\ z)$ has been replaced by the smaller element $(s_1-1,\ 1)$.

3) $s_1\neq 0$ and $z\neq 0$. Here we have

$$\tau((s_{|S|},\ \ldots,\ s_2,\ s_1-1,\ s_1),\ z-1)\ =$$
$$\{(s_{|S|}+1,\ 0),\ \ \ldots,\ \ (s_2+1,\ 0),\ \ (s_1,\ 0),\ \ (s_1,\ z-1)\}\ .$$

The element $(s_1,\ z)$ has been replaced by the two smaller elements $(s_1,\ 0)$ and $(s_1,\ z-1)$.

● *Solution* 2 : As our well-founded set, we take $\mathcal{M}(\mathbb{N})\times\mathbb{N}$, that is, the set of pairs where the first component is a multiset over the natural numbers and the second component is a natural number. For example, the pair $(\{3,\ 3,\ 0,\ 1,\ 1,\ 2\},\ 2)$ is an element of this set. The appropriate termination function at L is

$$\tau(S,\ z)\ =\ (\{s_i\in S\ :\ (\forall j<i)\ s_i\geq s_j\},\ \ z)\ ;$$

thus, the first component of the pair is a multiset containing those elements s_i in the stack S for which none of $s_{i-1},\ \ldots,\ s_2,\ s_1$ are larger than s_i . Note that s_1 always belongs to the multiset. For example, if $S=(3,\ 1,\ 0,\ 0,\ 1)$ and $z=0$, then $\tau(S,\ z)=(\{3,\ 1,\ 1\},\ 0)$. The same case analysis as in the previous solution applies. □

EXAMPLE 4: *Program schema for double recursion.*

Consider the following program schema that utilizes multisets:

```
$z$ := $e$
$S$ := $\{x\}$
loop $L$: assert $f(x)=h(f(s_1), h(\ldots, h(f(s_{|S|}), z)\ldots))$
          until $S=\{\}$
          $y$ :∈ $S$
          $S$ := $S \setminus \{y\}$
          if $p(y)$ then $z$ := $h(g(y), z)$
                    else $S$ := $S \cup \{k(y), l(y)\}$
                    fi
          repeat
assert $z=f(x)$ .
```

Here S is the multiset $\{s_1, s_2, \ldots, s_{|S|}\}$, $\{\}$ is the empty multiset, the statement $y{:}\in S$ is the nondeterministic assignment of an arbitrary element of S to y, and $S \setminus \{y\}$ is S with one occurrence of y removed. By instantiating the predicate variable p and the function variables h, g, k, l, and e, one obtains an instance of the schema that computes some particular function $f(x)$.

This iterative program computes the same function $f(x)$ as the recursive program schema

$$F(x) \Leftarrow \textbf{if } p(x) \textbf{ then } g(x) \textbf{ else } h(F(k(x)), F(l(x))) \textbf{ fi },$$

provided that the function h is associative and commutative and e is its identity element — i.e. for all u, v, and w, $h(u, h(v, w))=h(h(u, v), w)$, $h(u, v)=h(v, u)$, and $h(u, e)=u$.

We wish to show that the loop of the iterative program terminates for every instantiation, over some domain D, for which there exists a well-founded ordering $(D, \succ)$ such that

$$\neg p(x) \supset x \succ k(x) \wedge x \succ l(x) .$$

(It is under this condition that the recursive program terminates.) To prove termination, consider the multiset ordering $(\mathcal{M}(D), \gg)$ over the given domain $(D, \succ)$ and the termination function $\tau(S)=S$ at L. With each iteration an element y is either removed from S or replaced by the two smaller elements $k(y)$ and $l(y)$, thereby decreasing τ.

□

Remark: The previous examples suggest the following heuristic for proving termination: given a program over a domain $(D, \succ)$ that computes some function $f(x)$, if the program has a loop invariant of the form

$$f(x) = h(f(g_1(y)), f(g_2(y)), \ldots, f(g_n(y))) ,$$

try the multiset ordering $(\mathcal{M}(D), \gg)$, and use the termination function

$$\tau(y) = \{g_1(y), \quad g_2(y), \quad \ldots, \quad g_n(y)\} .$$

The idea underlying this heuristic is that τ represents the set of unevaluated arguments of some recursive definition of the function f. □

IV. TERMINATION OF PRODUCTION SYSTEMS

A *production system* Π over a set of expressions E is a (finite or infinite) set of rewriting rules, called *productions*, each of the form

$$\pi(\alpha,\ \beta,\ \ldots\ )\ \Rightarrow\ \pi'(\alpha,\ \beta,\ \ldots\ )\ ,$$

where α , β , ... are variables ranging over E . (The variables appearing in π' must be a subset of those in π .) Such a rule is applied in the following manner: given an expression $e \epsilon E$ that contains a subexpression

$$\pi(a,\ b,\ \ldots\ )\ ,$$

(i.e. the variables α, β, ... are instantiated with the expressions a, b, ... , respectively) replace that subexpression with the corresponding expression

$$\pi'(a,\ b,\ \ldots\ )\ .$$

We write $e \rightarrow e'$, if the expression e' can be derived from e by a single application of some rule in Π to one of the subexpressions of e .

For example, the following is a production system that differentiates an expression, containing $+$ and $\cdot$, with respect to x :

$$\boxed{\begin{array}{lll} Dx & \Rightarrow & 1 \\ Dy & \Rightarrow & 0 \\ D(\alpha+\beta) & \Rightarrow & (\ D\alpha + D\beta\ ) \\ D(\alpha\cdot\beta) & \Rightarrow & (\ (\beta\cdot D\alpha) + (\alpha\cdot D\beta)\ )\ , \end{array}}$$

where y can be any constant or any variable other than x . Consider the expression

$$D(D(x\cdot x)+y)\ .$$

We could either apply the third production to the outer D , or else we could apply the fourth production to the inner D . In the latter case, we obtain

$$D(((x\cdot Dx)+(x\cdot Dx))+y)\ ,$$

which now contains three occurrences of D . At this point, we can still apply the third production to the outer D , or we could apply the first production to either one of the inner D's . Applying the third production yields

$$(D((x\cdot Dx)+(x\cdot Dx))+Dy)\ .$$

Thus,

$$D(D(x\cdot x)+y)\ \rightarrow\ D(((x\cdot Dx)+(x\cdot Dx))+y)\ \rightarrow\ (D((x\cdot Dx)+(x\cdot Dx))+Dy)\ .$$

In general, at each stage in the computation there are many ways to proceed, and the choice is made nondeterministically. In our case, all choices eventually lead to the expression

$$((((1\cdot 1)+(x\cdot 0))+((1\cdot 1)+(x\cdot 0)))+0) \ ,$$

for which no further application of a production is possible.

A production system Π *terminates* over E, if there exist no infinite sequences of expressions $e_1, e_2, e_3, \ldots$ such that $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots$ and e_1 is an expression in E. In other words, given any initial expression, execution always reaches a state for which there is no way to continue applying productions. The difficulty in proving termination of a production system such as the one for differentiation above, stems from the fact that while some productions (the first two) may decrease the size of an expression, other productions (the last two) may increase its size. Also, a production (the fourth) may actually duplicate occurrences of subexpressions. Furthermore, applying a production to a subexpression, not only affects the structure of that subexpression, but also changes the structure of its superexpressions, including the top-level expression. And a proof of termination must take into consideration the many different possible sequences, generated by the nondeterministic choice of productions and subexpressions.

The following theorem has provided the basis for most of the techniques used for proving the termination of production systems:

> THEOREM: *A production system over E terminates, if and only if there exists a well-founded set* $(W, \succ)$ *and a termination function* $\tau : E \rightarrow W$, *such that for any* $e, e' \in E$
>
> $$e \rightarrow e' \text{ implies } \tau(e) \succ \tau(e') \ .$$

Proof:

- "only if" part. Assume that the system does always terminate, then the set E is well-founded under the $\overset{+}{\rightarrow}$ ordering, where $\overset{+}{\rightarrow}$ is the transitive closure of the relation $\rightarrow$. Let $(W, \succ)$ be $(E, \overset{+}{\rightarrow})$ and let τ be the identity function. Then clearly $e \rightarrow e'$ implies $\tau(e) = e \overset{+}{\rightarrow} e' = \tau(e')$.
- "if" part. Assume that $e \rightarrow e'$ implies $\tau(e) \succ \tau(e')$ in some well-founded set $(W, \succ)$. Suppose that the system does not terminate. Then by definition, for some expression $e \in E$, there exists an infinite sequence of expressions $e = e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots$. In that case, there exists an infinite decreasing sequence $\tau(e_1) \succ \tau(e_2) \succ \tau(e_3) \succ \ldots$ in W, which contradicts the assumption that $\succ$ is a well-founded ordering. Thus, it follows that the system must terminate. □

Several works have considered the problem of proving the termination of production systems. Among them: Gorn [1965] is an early work that addresses this issue; Iturriaga [1967] gives sufficient conditions under which a class of production systems terminates; Knuth and Bendix [1969] define a well-founded ordering based on a weighted size for expressions; Manna and Ness [1970] and Lankford [1975] use a "monotonic interpretation" that decreases with each application of a production; Lipton and Snyder [1977] make use of a "value-preserving" property as the basis for a method of proving termination.

In the following examples, we illustrate the use of multisets in proving termination alongside some previous methods.

EXAMPLE 1: *Associativity.*

Consider the set of arithmetic expressions E constructed from some set of atoms (symbols) and the single operator $+$. The production system

$$\boxed{(\alpha+\beta)+\gamma \quad \Rightarrow \quad \alpha+(\beta+\gamma)}$$

over E, contains just one production which reparenthesizes a sum by associating to the right. For example, the expression $(a+b)+((c+d)+e)$ becomes either $a+(b+((c+d)+e))$ or $(a+b)+(c+(d+e))$, both of which become $a+(b+(c+(d+e)))$. Since the size of the expression remains constant when the production is applied, some other measure is needed to prove termination.

● *Solution* 1 (Manna and Ness): Let the well-founded set be $(\mathbb{N}, >)$. The termination function $\tau{:}E\to\mathbb{N}$ maps expressions into the well-founded set, and is defined recursively as follows:

$$\tau(\alpha+\beta) \;=\; 2\cdot\tau(\alpha)+\tau(\beta)$$

for expressions of the form $\alpha+\beta$, and

$$\tau(u) \;=\; 1$$

for any atom u. For example, the value of τ for the expression $(a+b)+((c+d)+e)$ is $2\cdot(2\cdot1+1)+(2\cdot(2\cdot1+1)+1) = 13$.

The key point in the proof is that this function possesses the following two important properties:

1) The value of the termination function τ *decreases* for the subexpression that the production is applied to, i.e. for any possible value of α, β, and γ,

$$\tau((\alpha+\beta)+\gamma) \;>\; \tau(\alpha+(\beta+\gamma)) .$$

This is so since

$$\tau((\alpha+\beta)+\gamma) = 2\cdot\tau(\alpha+\beta)+\tau(\gamma) = 4\cdot\tau(\alpha)+2\cdot\tau(\beta)+\tau(\gamma) ,$$

while

$$\tau(\alpha+(\beta+\gamma)) = 2\cdot\tau(\alpha)+\tau(\beta+\gamma) = 2\cdot\tau(\alpha)+2\cdot\tau(\beta)+\tau(\gamma) ,$$

and $\tau(\alpha)$ is at least 1 .

2) The function τ is *monotonic in each operand* in the sense that if

$$\tau(e_1) > \tau(e_2) ,$$

for some expressions e_1 and e_2, then for any expression e_3,

$$\tau(e_1+e_3) > \tau(e_2+e_3) ,$$

and

$$\tau(e_3+e_1) > \tau(e_3+e_2) .$$

Thus, if $e \rightarrow e'$, for the outermost expression e, then some subexpression $(\alpha+\beta)+\gamma$ of e has been replaced by $\alpha+(\beta+\gamma)$ to obtain e'. We have $\tau((\alpha+\beta)+\gamma) > \tau(\alpha+(\beta+\gamma))$, by the first property. Therefore, by the monotonicity property, we get that

$$e \rightarrow e' \text{ implies } \tau(e) > \tau(e') ,$$

and by the theorem, it follows that the production system must terminate.

● *Solution* 2 (Knuth and Bendix): For this solution, the termination function $\tau(e)$ yields a sequence of natural numbers, listing the sizes of the subexpressions of e in *preorder*: the sequence begins with the size of e, $|e|$, and is followed by the sequence of sizes corresponding to the left operand of e, and then by the sequence of sizes corresponding to the right operand of e. These sequences of sizes are compared lexicographically. However, in order for a set of lexicographically-ordered sequences to be well-founded, the sequences must be of bounded length. In fact, the length of τ is constant for each computation, since the number of subexpressions is unchanged by the production.

To prove termination by this method, we need to show that any application of the production has the following two properties:

1) The value of the termination function τ *decreases* with each application of a production. Since

$$|(\alpha+\beta)+\gamma| = |\alpha+(\beta+\gamma)| ,$$

we proceed to compare the left operand $\alpha+\beta$ with α. But

$$|\alpha+\beta| > |\alpha| ,$$

and therefore $\tau(\pi)$ is lexicographically greater than $\tau(\pi')$.

2) Since the production does not change the size of the expression it is applied to, i.e.

$$|\pi| = |\pi'| ,$$

the sizes of all the expressions preceding π in the preorder are also unchanged.

Thus, $e \rightarrow e'$ implies that $\tau(e)$ is lexicographically greater than $\tau(e')$.

● *Solution* 3 (multisets): For this solution, we use the multiset ordering over the natural numbers, $(\mathcal{M}(\mathbb{N}), \gg)$, and let $\tau: E \rightarrow \mathcal{M}(\mathbb{N})$ return the multiset of the sizes $|\alpha|$ of all the subexpressions of the form $\alpha+\beta$ in e, i.e.

$$\tau(e) = \{|\alpha| : \alpha+\beta \text{ in } e\} .$$

For example,

$$\tau((a+b)+((c+d)+e)) = \{1, 3, 1, 3\} ,$$

since the left operands of the operator + are a, $a+b$, c, and $c+d$.

Again there are two crucial properties:

1) The value of the termination function τ *decreases* with each application of a production, i.e.

$$\tau((\alpha+\beta)+\gamma) \gg \tau(\alpha+(\beta+\gamma)) .$$

Before an application of the production, the multiset $\tau((\alpha+\beta)+\gamma)$ includes one occurrence of $|\alpha+\beta|$ and one of $|\alpha|$, along with elements corresponding to the subexpressions of α, β, and γ. After application of the production, the new multiset $\tau(\alpha+(\beta+\gamma))$ includes one occurrence of $|\alpha|$ and one of $|\beta|$, leaving the subexpressions of α, β, and γ unchanged. Thus, the element $|\alpha+\beta|$ has been replaced by the smaller element $|\beta|$, and the multiset has accordingly been decreased.

2) Since the production does not change the size of the expression it is applied to, i.e.

$$|\pi| = |\pi'| ,$$

the size of superexpressions containing $(\alpha+\beta)+\gamma$ is also unchanged.

The multiset $\tau(e)$ consists of all the elements in $\tau((\alpha+\beta)+\gamma)$ plus the sizes of the superexpressions and the sizes of their other subexpressions. The only elements in $\tau(e)$ that are changed by the production are those in $\tau(\pi)$, and they have been decreased by the production. Thus, $e \rightarrow e'$ implies that $\tau(e) \gg \tau(e')$. □

EXAMPLE 2: *Distributivity.*

The following system has two productions to apply the distributive law to an arithmetic expression composed of atoms and the operators + and • :

$$\boxed{\begin{array}{lll} \alpha\cdot(\beta+\gamma) & \Rightarrow & (\alpha\cdot\beta)+(\alpha\cdot\gamma) \\ (\beta+\gamma)\cdot\alpha & \Rightarrow & (\beta\cdot\alpha)+(\gamma\cdot\alpha)\ . \end{array}}$$

Both productions increase the size of the expression.

- *Solution* 1 (Manna ana Ness): Take $(\mathbb{N}, >)$ as the well-founded set. Let the termination function $\tau: E \rightarrow \mathbb{N}$ be defined by

$$\tau(\alpha+\beta) = \tau(\alpha)+\tau(\beta)+1$$

and

$$\tau(\alpha\cdot\beta) = \tau(\alpha)\cdot\tau(\beta)\ ,$$

for expressions of the form $\alpha+\beta$ or $\alpha\cdot\beta$, respectively, and let

$$\tau(u) = 2\ ,$$

for any atom u.

1) The value of the termination function τ decreases with each application of a production. In fact, both productions decrease τ from

$$\tau(\alpha)\cdot(\tau(\beta)+\tau(\gamma)+1) = \tau(\alpha)\cdot\tau(\beta) + \tau(\alpha)\cdot\tau(\gamma) + \tau(\alpha)$$

to

$$\tau(\alpha)\cdot\tau(\beta) + \tau(\alpha)\cdot\tau(\gamma) + 1\ .$$

Since $\tau(\alpha) \geq 2$, this is a decrease of at least 1.

2) The function τ is monotonic in each operand of + and •.

It follows that $e \rightarrow e'$ implies $\tau(e) > \tau(e')$.

- *Solution* 2 (Lipton and Snyder): For this method to be applicable, each production $\pi \Rightarrow \pi'$ must satisfy

$$|\pi| < |\pi'| ,$$

i.e. the size of the expression must be increased by the production, as is indeed the case in this example.

Now, consider the function $val2{:}E \rightarrow \mathbb{N}$, which maps expressions into the set of natural numbers, and returns the arithmetic value of the expression when the value 2 is assigned to each atom. For example, $val2((a+1)\cdot(c+0))=(2+2)\cdot(2+2)=16$. This function has the following two properties, which ensure termination:

1) $val2$ is *monotonic* in the sense that for any subexpression e' of an expression e,

$$val2(e) > val2(e') .$$

2) The productions are *value-preserving* for $val2$, i.e.

$$val2(\pi) = val2(\pi') ,$$

for each production $\pi \Rightarrow \pi'$.

Suppose that the system does not terminate. Then there exists an infinite sequence of expressions of the form $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots$. By the value-preserving property, $val2(e_1)=val2(e_2)=val2(e_3)= \ldots$. Furthermore, for any given value c, the monotonicity property clearly imposes a maximum depth c — and consequently a maximum size — on any expression e such that $val2(e)=c$. In particular, since the expressions in the sequence have a constant value, they have a maximum size, say m, i.e. $|e_i| \leq m$ for any e_i in the sequence. On the other hand, since each production increases the size of the subexpression it is applied to, it follows that $|e_1|<|e_2|<|e_3|< \ldots$, and, consequently, there must be some n such that $|e_n|>m$. But this is a contradiction. The system must therefore terminate.

- *Solution* 3 (multisets): For this solution, we use multisets over natural numbers, $(\mathcal{M}(\mathbb{N}), \gg)$. The termination function $\tau{:}E \rightarrow \mathcal{M}(\mathbb{N})$ is defined by

$$\tau(e) = \{val1(\alpha \cdot \beta) : \alpha \cdot \beta \text{ in } e\} ,$$

where $val1(\alpha \cdot \beta)$ gives the arithmetic value of $\alpha \cdot \beta$ when all the atoms are assigned the value 1.

1) The value of the termination function τ decreases with each application of a production. Applying the first production replaces the element $val1(\alpha\cdot(\beta+\gamma))$ in the multiset $\tau(\pi)$ with the two smaller elements $val1(\alpha\cdot\beta)$ and $val1(\alpha\cdot\gamma)$. The production also duplicates the products in α , but the value of each subexpression of α must also be less than the value of $\alpha\cdot(\beta+\gamma)$. Thus,

$$\tau(\pi) \gg \tau(\pi') .$$

The same is true for the second production.

2) Since

$$val1(\pi) = val1(\pi') ,$$

the value of superexpressions of π in the multiset $\tau(e)$ is preserved by the productions.

Therefore, $e \rightarrow e'$ implies $\tau(e) \gg \tau(e')$, and the system must terminate.

● *Solution 4* (nested multisets): Note that the products are reduced in size by each production. One would therefore like to prove termination using the well-founded set $(\mathcal{M}(\mathbb{N}), \gg)$, and a termination function that yields the multiset containing $|\alpha\cdot\beta|$ for each occurrence of a product $\alpha\cdot\beta$:

$$\tau(e) = \{|\alpha\cdot\beta| \ : \ \alpha\cdot\beta \text{ in } e\} .$$

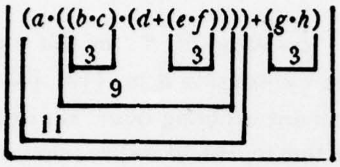
The value of this function is decreased by the application of a production, i.e. $\tau(\pi) \gg \tau(\pi')$ for each of the two productions $\pi \Rightarrow \pi'$. The problem is that the size of superexpressions increases, since $|\pi'| > |\pi|$; applying a production to a subexpression of e , will therefore increase $\tau(e)$.

To overcome this problem, we need a termination function that takes the nested structure of the expression into consideration, and gives more significance to more deeply nested products. Fortunately, this is exactly what nested multisets can do for us. Since this is the first time we illustrate the use of nested multisets, we shall discuss this solution in greater detail.

Let the well-founded set be the nested multisets over the natural numbers, $(\mathcal{M}^*(\mathbb{N}), \gg^*)$, and let the termination function $\tau : E \rightarrow \mathcal{M}^*(\mathbb{N})$ yield $|\alpha\cdot\beta|$ for each occurrence of a product $\alpha\cdot\beta$, while preserving the nested structure of the expression. For example, the expression $(a\cdot((b\cdot c)\cdot(d+(e\cdot f))))+(g\cdot h)$ contains five subexpressions of form $\alpha\cdot\beta$. Their sizes are:

$$|a\cdot((b\cdot c)\cdot(d+(e\cdot f)))| = 11 ,$$
$$|(b\cdot c)\cdot(d+(e\cdot f))| = 9 ,$$
$$|b\cdot c| = 3 ,$$
$$|e\cdot f| = 3 , \text{ and}$$
$$|g\cdot h| = 3 .$$

Considering the nested depths of the products, the structure of the expression is



Thus, for

$$e = (\ (a\ \cdot\ (\ (b\cdot c)\ \cdot\ (d+(e\cdot f))\ )\ )\ +\ (g\cdot h)\ )\ ,$$

we obtain

$$\tau(e) = \{\ \{11,\ \{\{3\}, 9,\ \{3\}\}\},\qquad \{3\}\ \}\ .$$

1) For each production $\pi\Rightarrow\pi'$, we have

$$\tau(\pi) \gg^* \tau(\pi') ,$$

under the nested multiset ordering. We have

$$\tau(\alpha\cdot(\beta+\gamma)) = \{\ \{|\alpha\cdot(\beta+\gamma)|,\ \overline{\tau(\alpha)},\ \overline{\tau(\beta)},\ \overline{\tau(\gamma)}\}\ \}$$

while

$$\tau((\alpha\cdot\beta)+(\alpha\cdot\gamma)) = \{\ \{|\alpha\cdot\beta|,\ \overline{\tau(\alpha)},\ \overline{\tau(\beta)}\},\quad \{|\alpha\cdot\gamma|,\ \overline{\tau(\alpha)},\ \overline{\tau(\gamma)}\}\ \}\ ,$$

where $\overline{\tau(\alpha)}$, $\overline{\tau(\beta)}$, and $\overline{\tau(\gamma)}$ stand for the elements of the multisets $\tau(\alpha)$, $\tau(\beta)$, and $\tau(\gamma)$, respectively. This is a decrease in $\mathcal{M}^*(\mathbb{N})$, since $\{|\alpha\cdot(\beta+\gamma)|,\ \overline{\tau(\alpha)},\ \overline{\tau(\beta)},\ \overline{\tau(\gamma)}\}$ is greater than both $\{|\alpha\cdot\beta|,\ \overline{\tau(\alpha)},\ \overline{\tau(\beta)}\}$ and $\{|\alpha\cdot\gamma|,\ \overline{\tau(\alpha)},\ \overline{\tau(\gamma)}\}$, regardless of the exact form of α, β and γ.

For example,

$$\tau((b\cdot c)\cdot(d+(e\cdot f))) = \{\{\{3\},\ 9,\ \{3\}\}\}\ ,$$

and applying the first production yields

$$\tau(((b\cdot c)\cdot d)+((b\cdot c)\cdot(e\cdot f))). = \{\{\{3\},\ 5\},\ \{\{3\},\ 7,\ \{3\}\}\}\ .$$

This is a decrease in the nested multiset ordering, since $\{\{3\},\ 9,\ \{3\}\}$ is greater than both $\{\{3\},\ 5\}$ and $\{\{3\},\ 7,\ \{3\}\}$. A similar argument applies to the other production.

2) It remains to ascertain what happens to the value of τ for superexpressions. The crucial point here is that the termination function gives greater weight to the more deeply nested products by placing their size at a greater depth in the nested multiset. The effect of the productions on lower-level expressions is therefore more significant than their effect on higher-level expressions, and the decrease in τ for the subexpression to which the production is applied, overshadows any increase in the size of a superexpression.

Consider, for example, $a\cdot(b\cdot(c+d)) \rightarrow a\cdot((b\cdot c)+(b\cdot d))$. The value of τ for the expression on the left is $\{\{7, \{5\}\}\}$, while for the right-hand side expression it is $\{\{9, \{3\}, \{3\}\}\}$. Note that this represents a decrease in the nested multiset ordering over $\mathbb{N}$, despite the fact that the element 7 , corresponding to the size of the top-level expression, has been increased to 9 . This is the case since the production has replaced the element $\{5\}$ in the multiset by two occurrences of the smaller $\{3\}$, and $\{5\}$ is also greater than 9 — or any number for that matter — on account of its greater depth.

Thus, $e \rightarrow e'$ implies $\tau(e) \gg^* \tau(e')$. □

EXAMPLE 3: *Differentiation.*

The following system symbolically differentiates an expression with respect to x :

$$\boxed{\begin{array}{lcl}
Dx & \Rightarrow & 1 \\
Dy & \Rightarrow & 0 \\
D(\alpha+\beta) & \Rightarrow & (\ D\alpha + D\beta\ ) \\
D(\alpha\cdot\beta) & \Rightarrow & (\ (\beta\cdot D\alpha) + (\alpha\cdot D\beta)\ )\ , \\
D(-\alpha) & \Rightarrow & (\ -D\alpha\ ) \\
D(\alpha-\beta) & \Rightarrow & (\ D\alpha - D\beta\ ) \\
D(\alpha/\beta) & \Rightarrow & (\ (D\alpha/\beta) - ((\alpha\cdot D\beta)/(\beta\uparrow 2))\ ) \\
D(ln\ \ \alpha) & \Rightarrow & (\ D\alpha\ /\ \alpha\ ) \\
D(\alpha\uparrow\beta) & \Rightarrow & (\ (D\alpha\cdot(\beta\cdot(\alpha\uparrow(\beta-1)))) + (((ln\ \ \alpha)\cdot D\beta)\cdot(\alpha\uparrow\beta))\ )\ .
\end{array}}$$

● *Solution* 1 (Manna and Ness): Take $(\mathbb{N}, >)$ as the well-founded set. Let the termination function $\tau{:}E \rightarrow \mathbb{N}$ be defined by

$$\tau(\alpha\otimes\beta) = \tau(\alpha)+\tau(\beta)\ ,$$

where $\otimes$ is any of the binary operators $+$, $\cdot$, $-$, and $\uparrow$,

$$\tau(D\alpha) = \tau(\alpha)^2 ,$$
$$\tau(-\alpha) = \tau(\alpha)+1 ,$$
$$\tau(ln\ \alpha) = \tau(\alpha)+1 ,$$

and

$$\tau(u) = 4 ,$$

for any atom u. (If the last three productions were not included in the system, then $\tau(u)=2$ would suffice.)

1) For each of the nine productions $\pi \Rightarrow \pi'$, the value of τ decreases, i.e. $\tau(\pi) > \tau(\pi')$. For example,

$$\tau(D(\alpha/\beta)) = (\tau(\alpha)+\tau(\beta))^2 = \tau(\alpha)^2 + \tau(\beta)^2 + 2\cdot\tau(\alpha)\cdot\tau(\beta) ,$$

while

$$\tau(((D\alpha/\beta)-((\alpha\cdot D\beta)/(\beta\uparrow 2)))) = \tau(\alpha)^2 + \tau(\beta)^2 + \tau(\alpha) + 2\cdot\tau(\beta) + 4 .$$

This is a decrease, since $2\cdot\tau(\alpha)\cdot\tau(\beta) \geq 4\cdot\tau(\alpha)+4\cdot\tau(\beta) > \tau(\alpha)+2\cdot\tau(\beta)+4$.

2) τ is monotonic in each operand.

It follows that $e \rightarrow e'$ implies $\tau(e) > \tau(e')$.

● *Solution* 2 (multisets): To prove termination, we use the multiset ordering over sequences of natural numbers. The sequences are compared under the ***stepped lexicographic*** order $\succ$, i.e. longer sequences are greater than shorter ones (regardless of the values of the individual elements), and equal length sequences are compared lexicographically. The termination function is

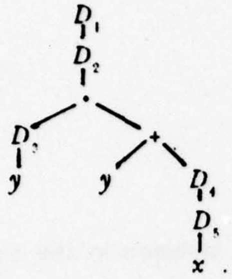
$$\tau(e) = \{(d_1(x),\ d_2(x),\ \ldots) : x \text{ is an occurrence of an atom in } e\} ,$$

where $d_i(x)$ is the distance (number of operators) between x and the ith enclosing D.

For example, consider the expression

$$e=DD(Dy\cdot(y+DDx)) ,$$

or in tree form (with the D's enumerated for expository purposes),



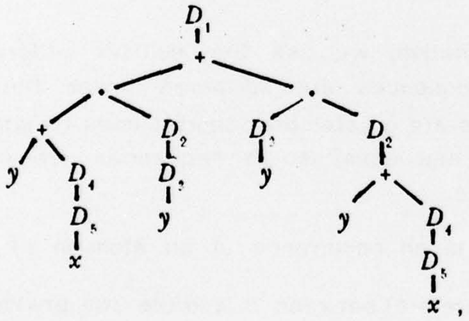
There are three atoms: y, y, and x. The left atom y contributes the element $(0, 2, 3)$ to the multiset, since there are no operators between D_3 and y, there are two operators ($\cdot$ and D_3) between D_2 and y, and there are three operators (D_2, $\cdot$, and D_3) between D_1 and y. Similarly the other two atoms contribute $(2, 3)$ and $(0, 1, 4, 5)$. Thus,

$$T(e) = \{ (0, 2, 3),\ (2, 3),\ (0, 1, 4, 5) \} .$$

Applying the production

$$D(\alpha \cdot \beta) \Rightarrow ((\beta \cdot D\alpha) + (\alpha \cdot D\beta)) ,$$

to e, yields $e' = D(((y+DDx)\cdot DDy)+(Dy\cdot D(y+DDx)))$. In tree form (with the labelling of the D's retained), we have



and accordingly

$$T(e') = \{ (3),\ (0, 1, 5),\ (0, 1, 4),\ (0, 3),\ (1, 4),\ (0, 1, 3, 6) \} ,$$

Thus, $T(e) \gg T(e')$, since the element $(0, 1, 4, 5)$ has been replaced by five shorter sequences and by the lexicographically smaller $(0, 1, 3, 6)$.

In general, the following two properties hold:

1) Applying any of the productions decreases T. Consider, for example, what happens to the multiset $T(e)$ when the production

$$D(\alpha \cdot \beta) \quad \Rightarrow \quad ((\beta \cdot D\alpha) + (\alpha \cdot D\beta))$$

is applied to some subexpression of e. Let x be an atom occurring in α. Applying the production results in replacing the sequence $s=(d_1(x),\ d_2(x),\ \ldots)$ corresponding to x, with two sequences, s' and s'', corresponding to the occurrences of x in $D\alpha$ and α, respectively. But s is greater than both s' and s'': the sequence s'' is shorter than s, since there is one less D above x; the sequence s' is of the same length as s, but is !exicographically less, since a D has been pushed closer to x, while the distance to nearer D's remains unchanged. Similarly, the sequences corresponding to the atoms in β are replaced by two smaller sequences.

2) The productions only affect the sequences in $T(e)$ corresponding to the atoms of the subexpression that they are applied to.

Therefore, for any application of a production, $e \rightarrow e'$ implies $T(e) \gg T(e')$.

● *Solution* 3 (nested multisets): Since the arguments to D are reduced in size by each production, and none of the productions increase the nested depth of D's, nested multisets constructed from the sizes of the arguments of D are an appropriate tool.

Let the well-founded set be the nested multisets over the natural numbers, $(\mathcal{M}^*(\mathbb{N}),\ \gg^*)$, and let the termination function $T{:}E \rightarrow \mathcal{M}^*(\mathbb{N})$ yield $|\alpha|$ for each occurrence of $D\alpha$, while preserving the nested structure of the expression. For example, the arguments of the six occurrences of D in the expression $D(D(Dx \cdot Dy)+Dy)/Dx$ are $D(Dx \cdot Dy)+Dy$, $Dx \cdot Dy$, x, y, y, and x. They are of sizes 9, 5, 1, 1, 1, and 1, respectively. Thus, for

$$e = D(D(Dx \cdot Dy) + Dy) / Dx) ,$$

we have

$$T(e) = \{\{9, \{5, \{1\}, \{1\}\}, \{1\}\}, \{1\}\} .$$

1) For each production $\pi \Rightarrow \pi'$, we have $T(\pi) \gg^* T(\pi')$. Consider, for example, the production

$$D(\alpha \cdot \beta) \quad \Rightarrow \quad ((\beta \cdot D\alpha) + (\alpha \cdot D\beta)) ,$$

and let $\overline{\tau(\alpha)}$ and $\overline{\tau(\beta)}$ stand for the list of elements of the multisets $\tau(\alpha)$ and $\tau(\beta)$, respectively. Applying τ to the two sides of the production, yields

$$\tau(D(\alpha\cdot\beta)) = \{\ \{|\alpha\cdot\beta|,\ \overline{\tau(\alpha)},\ \overline{\tau(\beta)}\}\ \}$$

and

$$\tau((\beta\cdot D\alpha)+(\alpha\cdot D\beta)) = \{\ \overline{\tau(\beta)},\ \ \{|\alpha|,\ \overline{\tau(\alpha)}\},\ \ \overline{\tau(\alpha)},\ \ \{|\beta|,\ \overline{\tau(\beta)}\}\ \}\ .$$

This clearly is a decrease in $\mathcal{M}^*(\mathbb{N})$, regardless of the exact form of α and β, since $\{|\alpha\cdot\beta|,\ \overline{\tau(\alpha)},\ \overline{\tau(\beta)}\}$ is greater than $\{|\alpha|,\ \overline{\tau(\alpha)}\}$ and $\{|\beta|,\ \overline{\tau(\beta)}\}$, and is also greater than each of the elements in $\overline{\tau(\alpha)}$ and $\overline{\tau(\beta)}$.

For example,

$$\tau(D(x\cdot Dy)) = \{\{4,\ \{1\}\}\}\ ,$$

while

$$\tau((x\cdot DDy)+(Dy\cdot Dx))) = \{\{2,\ \{1\}\},\ \{1\},\ \{1\}\}\ .$$

This is a decrease in the nested multiset order, since $\{4,\ \{1\}\}$ is greater than both $\{2,\ \{1\}\}$ and $\{1\}$. A similar argument applies to all of the other productions.

2) As in the previous example, the decrease in τ for the lower-level expression overshadows any increase in the size of a higher-level expression.

It follows that $e \rightarrow e'$ implies $\tau(e) \gg^* \tau(e')$. □

In this section, we have illustrated the use of multiset and nested multiset orderings in proofs of termination of production systems, by means of a number of examples. Along similar lines, using these orderings, one can give general theorems which express sufficient conditions for the termination of broad classes of production systems.

ACKNOWLEDGEMENT

We thank Bob Boyer, John Doner, Chris Goad, John McCarthy, Steve Ness, Amir Pnueli, Adir Pridor and Richard Weyhrauch for stimulating discussions.

REFERENCES

Dijkstra, E.W. [1976], *A small note on the additive composition of variant functions*, Note EWD592, Burroughs Corp., Nuenen, The Netherlands.

Floyd, R.W. [1967], *Assigning meanings to programs*, Proc. Symp. in Applied Mathematics, vol. 19 (J.T. Schwartz, ed.), American Mathematical Society, Providence, RI, pp. 19-32.

Gentzen, G. [1938], *New version of the consistency proof for elementary number theory*, in *The collected papers of Gerhart Gentzen* (M.E. Szabo, ed.), North Holland, Amsterdam (1969), pp. 252-286.

Gorn, S. [Sept. 1965], *Explicit definitions and linguistic dominoes*, Proc. Conf. on Systems and Computer Science, London, Ontario, pp. 77-115.

Iturriaga, R. [May 1967], *Contributions to mechanical mathematics*, Ph.D. thesis, Carnegie-Mellon Univ., Pittsburgh, PA.

Katz, S.M. and Z. Manna [1975], *A closer look at termination*, Acta Informatica, vol. 5, no. 4, pp. 333-352.

Knuth, D.E. and P.B. Bendix [1969], *Simple word problems in universal algebras*, in Computational Problems in Universal Algebras (J. Leech, ed.), Pergamon Press, Oxford, pp. 263-297.

Lankford, D.S. [Dec. 1975], *Canonical inference*, Memo ATP-32, Automatic Theorem Proving Project, Univ. of Texas, Austin, TX.

Lipton, R.J. and L. Snyder [Aug. 1977], *On the halting of tree replacement systems*, Proc. Conf. on Theoretical Computer Science, Waterloo, Canada, pp. 43-46.

Manna, Z. and S. Ness [Jan. 1970], *On the termination of Markov algorithms*, Proc. Third Hawaii Intl. Conf. on System Sciences, Honolulu, HI, pp. 789-792.

Manna, Z. and R.J. Waldinger [Feb. 1978], *Is SOMETIME sometimes better than ALWAYS? Intermittent assertions in proving program correctness*, CACM, vol. 21, no. 2, pp. 159-172.